Information Sciences 648 (2023) 119580

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

HiSTGNN: Hierarchical spatio-temporal graph neural network for weather forecasting

Minbo Ma [a], Peng Xie [a], Fei Teng [a], Bin Wang [b], Shenggong Ji [c], Junbo Zhang [d], Tianrui Li [a,*]

[a] School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, 611756, China
[b] School of Computer Science and Technology, Ocean University of China, Qingdao, 266100, China
[c] Tencent Inc., Shenzhen, 518000, China
[d] JD iCity, JD Technology, JD Intelligent Cities Research, Beijing, 100000, China

ARTICLE INFO

Keywords:
Weather forecasting
Spatio-temporal forecasting
Hierarchical graph neural network

ABSTRACT

Weather forecasting is an attractive yet challenging task due to its significant impacts on human life and the intricate nature of atmospheric motion. Deep learning-based techniques, utilizing abundant observations, have gained popularity in recent years. However, many existing methods mainly explore temporal patterns of meteorological variables while neglecting the interactions between variables across regions. To address this limitation, we propose HiSTGNN, a novel Hierarchical Spatio-temporal Graph Neural Network, which enables accurate predictions of multiple variables and stations over multiple time steps. HiSTGNN incorporates an adaptive graph learning module that constructs a self-learning hierarchical graph, comprising a global graph representing regions and a local graph capturing meteorological variables for each region. By leveraging graph convolution and gated temporal convolution with a dilated inception as the backbone, we effectively capture hidden spatial dependencies and diverse long-term meteorological trends. Additionally, we introduce dynamic interactive learning to facilitate bidirectional information between the two-level graphs. Experiments on three real-world meteorological datasets demonstrate the superiority of our proposed method compared to seven well-known baselines, including convolutional neural networks, recurrent neural networks, and spatio-temporal forecasting methods. Notably, HiSTGNN achieves remarkable improvements over the state-of-the-art weather forecasting method on the WD_BJ dataset, with a 4.25% decrease in MAE and a 5.34% decrease in RMSE.

1. Introduction

Weather forecasting plays a profound role in various aspects of human livelihood, enabling informed decision-making in agriculture, transportation, energy, and ensuring public safety [1]. The discipline has evolved significantly since its early beginnings when Robert FitzRoy, a British naval officer in the 19th century, utilized telegraph lines to gather data on atmospheric pressure and wind direction [2]. Today, weather forecasting has transformed into a sophisticated field that relies on diverse data sources and

* Corresponding author.
E-mail addresses: minboma@my.swjtu.edu.cn (M. Ma), pengxie@my.swjtu.edu.cn (P. Xie), fteng@swjtu.edu.cn (F. Teng), wangbin9545@ouc.edu.cn (B. Wang), shenggongji@163.com (S. Ji), msjunbozhang@outlook.com (J. Zhang), trli@swjtu.edu.cn (T. Li).

https://doi.org/10.1016/j.ins.2023.119580
Received 10 May 2023; Received in revised form 26 July 2023; Accepted 15 August 2023

Available online 21 August 2023

computational models to provide precise and reliable predictions. Despite the predominance of numerical weather prediction (NWP), limitations arising from the existing theory of atmospheric physics and challenges in solving differential equations can contribute to inaccurate predictions [3].

In the era of big data, data-driven methods have garnered considerable attention, with deep learning-based techniques emerging as prominent solutions. In particular, deep learning-based techniques, known for their ability to extract high-dimensional representations from historical observations, have exhibited remarkable computational efficiency and forecasting [4]. This study focuses on weather forecasting using data collected from ground weather stations. Specifically, we address the 3M task, which involves simultaneous predictions of multiple meteorological variables for multiple weather stations over multiple future time steps.

Existing studies approach this problem by treating it as time series forecasting and spatio-temporal forecasting tasks. The former focuses on analyzing and capturing the temporal dependencies of meteorological variables within a single selected ground weather station [5–7]. Meanwhile, the latter takes into account the spatial correlation of meteorological variables across multiple regions [8–10]. However, these studies still have limitations in capturing the intricate interdependencies within the Earth's atmospheric system. They fail to effectively capture the spatio-temporal interactions between meteorological variables at both local and regional scales.

Crucially, the interactions between meteorological variables play a crucial role in weather patterns and forecasting. For instance, temperature and humidity exhibit a strong mutual influence. Warmer air can hold more moisture, leading to higher humidity levels. Conversely, as humidity increases, it affects the rate of temperature change and can influence cloud formation and precipitation patterns [11]. These correlations extend beyond specific locations and encompass wider regions due to the dynamic nature of the atmosphere. Unraveling these intricate interactions and capturing the nonlinear temporal dynamics is crucial for achieving accurate weather forecasting, albeit it poses greater challenges.

To handle these challenges, we propose a Hierarchical Spatio-temporal Graph Neural Network for weather forecasting, named HiSTGNN. HiSTGNN employs a hierarchical graph to capture the interactions between meteorological variables across different regions. The hierarchical graph consists of a global graph and local graphs, where each node in the local graph represents a meteorological variable observed from a single weather station, and edges represent the correlations between variables. The global graph comprises nodes corresponding to the local graphs, facilitating information transfer between regions. To construct the hierarchical graph, we introduce an adaptive graph learning module to learn the structure from data. Furthermore, we design a spatio-temporal learning module based on the dilated inception network and graph convolutional network. This module analyzes a sequence of hierarchical graph snapshots, forming a hierarchical spatio-temporal graph, to capture long-term spatio-temporal dependencies across variables and regions. To enable bidirectional information flow between the two-level graphs, we introduce a dynamic interactive learning module. This module aggregates the representation of a local graph into the corresponding node in the global graph and facilitates information diffusion in the opposite direction.

HiSTGNN follows an end-to-end approach, optimizing all parameters through gradient descent. In summary, the contributions of this paper can be outlined as follows:

- This study pioneers the exploration of a hierarchical graph-based perspective combined with graph neural networks (GNNs) for deep learning in weather forecasting.
- We propose an adaptive hierarchical graph learning module that effectively captures hidden correlations among meteorological variables in different regions. Importantly, our method avoids the need for explicit graph structures guided by domain knowledge.
- Our novel HiSTGNN enables end-to-end weather forecasting, simultaneously learning feature representations and graph structures from weather data within an iterative framework.
- Extensive evaluations of HiSTGNN on three real-world weather datasets confirm its effectiveness, surpassing seven baselines, including convolutional neural networks (CNNs)/recurrent neural networks (RNNs)/GNNs and traditional time series methods.

2. Related work

Related work in this paper can be classified into two categories: weather forecasting and hierarchical graph neural networks.

2.1. Weather forecasting

Weather forecasting plays an important role in human livelihood, aiming to provide accurate and timely predictions of weather status. This technique can be divided into three categories based on underlying methodologies: Numerical Weather Forecasting, Traditional Machine Learning, and Deep Learning. Numerical Weather Forecasting models [12,3] utilize complex mathematical equations to simulate the physical processes of the atmosphere. By dividing the atmosphere into a three-dimensional grid, these models employ various differential equations to forecast the evolution of meteorological variables over time. This method tends to suffer from the issues of substantial computing resources and unstable modeling under inappropriate initial solutions. Recent research focuses on boosting prediction and post-processing to improve the reliability of weather forecasting [13,14]. Traditional Machine Learning and Deep Learning are data-driven methods that exploit meteorological patterns from historical weather data to learn an input-output mapping function, where the input is past weather observation data and the output is the weather forecast. Traditional machine learning treats weather forecasting as time series forecasting [4]. Autoregressive integrated moving average (ARIMA) [15,16] models the linear temporal dependency within weather data by combining autoregressive, differencing, and moving average components. Support vector regression (SVR) [17] uses kernel functions to transform weather data into a high-dimensional

Table 1
Frequently used notations.

Notation	Description
S	number of weather stations
M	number of meteorological variables
d	dimensionality of features
P	number of input time steps
Q	number of forecasting time steps
$x_t \in \mathbb{R}^{S\times M}$	observed values of M meteorological variables in S weather stations at time t
$\mathcal{G}$	graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$ with node set $\mathcal{V}$, edge set $\mathcal{E}$
$\mathcal{G}_H, \mathcal{G}_G, \mathcal{G}_L$	hierarchical graph, global graph and local graph
$\mathcal{A} \in \mathbb{R}^{N\times N}$	adjacency matrix of graph with N nodes
$\mathcal{H}^l$	hidden feature of HiSTGNN at lth layer

space to find a hyperplane that minimizes the prediction error. Recently, deep learning methods have shown a significant advantage in weather forecasting. DUQ [5] utilizes an encoder-decoder framework with RNNs and employs a negative log-likelihood error loss function for point forecasting and uncertainty quantification. CNNs-based methods [6,7,18] utilize dilated convolution and causal convolution to capture the various and long-term temporal dependencies. However, these purely temporal modeling methods lack spatial interactions between variables and between regions. For capturing spatio-temporal dependencies, Deep hybrid model [19] uses an ensemble of boosted decision-tree learners and spatial interpolation with a deep belief network [20] to respect temporal and spatial dependencies among weather variables. ConvLSTM [8] treats precipitation nowcasting as a spatio-temporal forecasting problem and combines CNNs and LSTMs to capture spatio-temporal hidden correlations in radar echo data simultaneously. A hybrid neural model [21] combining CNNs, RNNs, and attention mechanism, is utilized to extract the spatial and temporal correlation features for wind speed forecasting. More recently, many studies focus on applying spatio-temporal graph neural networks in spatio-temporal forecasting tasks like traffic prediction [22], taxi demand prediction [23], driver maneuver anticipation [24] and air quality analysis [25], in which GNNs is to capture hidden spatial dependency with graph structures representing relationships between variables and RNNs or CNNs is to model hidden temporal dependency. In the field of weather forecasting, various studies have explored the application of GNNs. For instance, InstGCNs [26] using RNNs and GNNs to cope with inherent nonlinearity and spatio-temporal correlation in the weather radar data improves Short-term quantitative precipitation forecasting. GE-STDGN [27] combines graph convolution and evolutionary multi-objective optimization to improve spatio-temporal weather forecasting. MasterGNN [28] adopts multi-task learning with a heterogeneous graph neural network for air quality and weather predictions. Additionally, Global weather forecasting method [29] utilizes GNNs to aggregate information on the sphere over a physically-uniform neighborhood of latitude-longitude grid. DeepSphere [9] and CLCRN [10] also introduce graph convolution on spherical weather data. GraphCast [30] employs GNNs to achieve message passing on a multi-mesh graph representation derived from the mapping of latitude-longitude grid weather observations for medium-range weather forecasting. Notably, these researches on the application of GNNs for weather forecasting focus on spatial relationship modeling between regions while neglecting the interplay between regions and meteorological variables.

2.2. Hierarchical graph neural networks

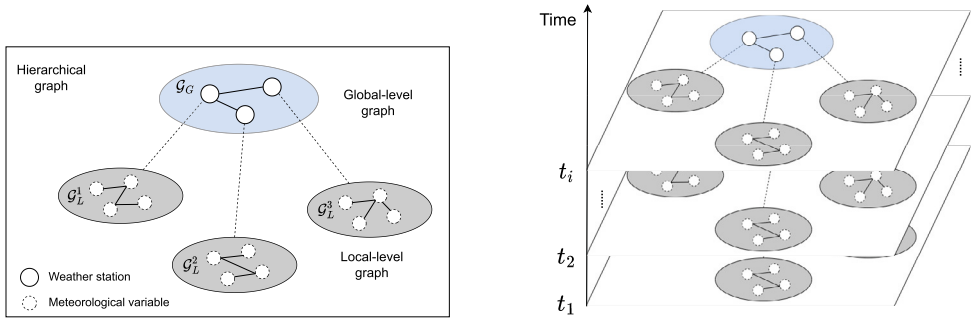
Existing GNNs encounter a significant constraint due to their flat architectures, which restricts their ability to aggregate information hierarchically. In response, a hierarchical graph neural network [31] was introduced for node classification tasks, generating hierarchical representations of a graph through pooling operations. Additionally, researchers in the field of traffic forecasting have explored hierarchical graph convolution networks, treating road networks and regional networks as a hierarchical graph to model spatial correlations [32]. However, these approaches exhibit limitations in information transfer, lacking bidirectional communication from top to bottom or bottom to top, which may result in information loss. In contrast to prior methods, our approach tackles these limitations by leveraging adaptive graph learning in an end-to-end framework. Our method establishes bidirectional information passing between the variable-level graph and the station-level graph. This comprehensive information exchange ensures the preservation and exchange of valuable insights between different graph levels.

3. Problem formulation

In this paper, we present the formalization of weather forecasting as a spatio-temporal forecasting task that involves predicting multiple meteorological variables for multiple weather stations over multiple future time steps. The mathematical notation used to describe our method is summarized in Table 1.

3.1. Definitions

Definition 1 *(Meteorological observations)*. Consider S spatially distributed ground weather stations, $x_t \in \mathbb{R}^{S\times M}$ denotes the observation of M meteorological variables at time step t, where the meteorological variables are the state of the atmosphere and the weather conditions in the corresponding regions.



(a) Hierarchical graph (b) Hierarchical spatio-temporal graph

Fig. 1. Hierarchical graph modeling the correlations between meteorological variables in different weather stations (i.e., regions).

Definition 2 *(Graph).* Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph, where $\mathcal{V}$ is a set of N nodes and $\mathcal{E}$ is a set of edges. The edges measure the correlations between nodes and can be mathematically expressed as an adjacent matrix $\mathcal{A} \in \mathbb{R}^{N \times N}$, where $\mathcal{A}_{i,j}$ denotes the correlation between nodes v_i and v_j.

Definition 3 *(Hierarchical graph).* Let $\mathcal{G}_H = \{\mathcal{G}_G, \{\mathcal{G}_L^i\}_{i=1}^S\}$ denote a hierarchical graph, as shown in Fig. 1(a), where $\mathcal{G}_G$ denotes the global graph and $\mathcal{G}_L$ denotes the local graph. We use $\mathcal{G}_G$ to represent the correlations between weather stations and use $\mathcal{G}_L^i$ to represent the correlations between meteorological variables that are observed in ith weather station. In the following, we use G and L as subscripts to distinguish global and local graphs.

Definition 4 *(Hierarchical spatio-temporal graph).* Let $\{\mathcal{G}_H^{t_i}\}_{i=1}$ denote a series of snapshots of the hierarchical spatio-temporal graph, as shown Fig. 1(b), which represents the consecutive changes of multiple meteorological variables in multiple regions.

3.2. Problem statement

Given historical meteorological data over P time steps $\mathcal{X} = \{x_{t_1}, x_{t_2}, \cdots, x_{t_P}\} \in \mathbb{R}^{S \times M \times P}$, our goal is to predict the states of multiple meteorological variables for all weather stations over the next consecutive P time steps $\hat{\mathcal{Y}} = \{\hat{x}_{t_{P+1}}, \hat{x}_{t_{P+2}}, \cdots, \hat{x}_{t_{P+Q}}\}$. More generally, the auxiliary features also can be coupled with the observations, such as station ID and time of the day. Assuming the inputs $\mathcal{X} \in \mathbb{R}^{S \times M \times P \times d}$, the weather forecasting problem is defined as follows,

$$\{\hat{x}_{t_{P+1}}, \hat{x}_{t_{P+2}}, \cdots, \hat{x}_{t_{P+Q}}\} = \mathcal{F}(\{x_{t_1}, x_{t_2}, \cdots, x_{t_P}\}), \tag{1}$$

where $\mathcal{F}$ is the mapping function from $\mathcal{X}$ to $\mathcal{Y}$ we aim to learn.

4. Methodology

In this section, we begin by presenting the overarching architecture of HiSTGNN. Subsequently, we provide a detailed description of each individual component.

4.1. Overall architecture

Fig. 2 visually represents the hierarchical and iterative architecture of HiSTGNN, which is specifically designed to extract spatio-temporal features from meteorological variables and regions. The framework is comprised of three main components: *adaptive graph learning module* (AGL), *spatio-temporal learning module* (STL), and *dynamic interactive learning module* (DIL). To discover the implicit correlations between meteorological variables and weather stations, AGL module constructs self-learning local graphs and a global graph in the form of graph adjacency matrices, which are later fed into graph neural networks. STL module consists of a graph convolutional network and a temporal convolutional network with dilated inception. This combination effectively captures spatial and temporal dependencies within the data. To build the bidirectional information passing between the two-level graphs, DIL module is divided into *information fusion layer* and *information diffusion layer*, which are interleaved with the spatio-temporal learning module of local graphs and global graph. Further details of HiSTGNN are presented in the subsequent sections.

4.2. Adaptive graph learning module

AGL module learns a hierarchical graph from observed data to capture the hidden dependencies between weather variables both locally and across multiple regions. The construction of such a hierarchical graph depends on assessing the correlation between nodes within variable-level graphs (i.e., local graphs), and nodes within the station-level graph (i.e., the global graph). Existing

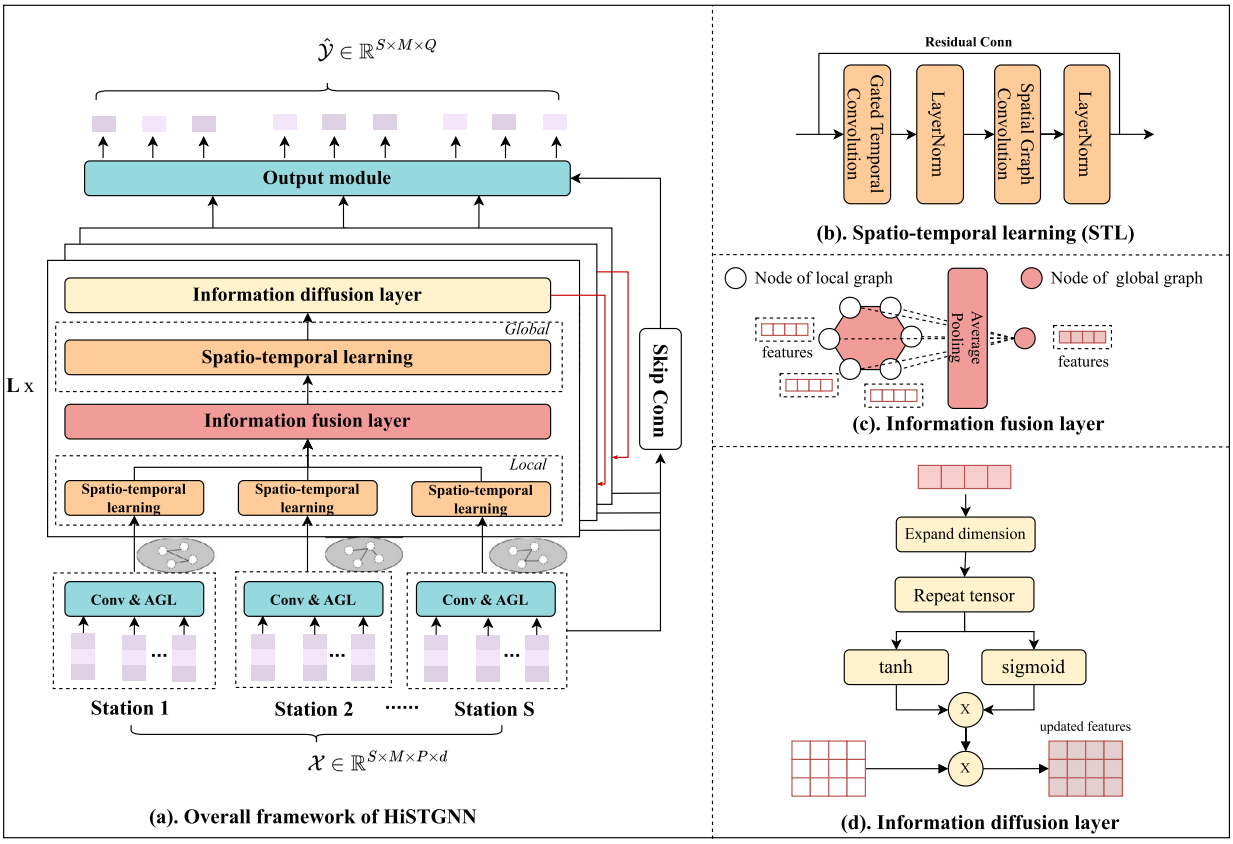


Fig. 2. The overall framework of HiSTGNN. The Conv is a 1×1 standard convolution to project the inputs to latent space. AGL is the adaptive graph learning module to generate the adjacency matrix of both local graphs and the global graph. STL is the spatio-temporal learning module to capture spatial and temporal dependencies. The information fusion layer and information diffusion layer are used to build interactions between two-level graphs. The output module is a 1×1 standard convolution to output desired meteorological variables and future time steps. The hyper-parameter L represents the number of stacked layers.

studies mainly measure the correlation of weather variables between regions by Euclidean distance [10,28]. However, this may restrict the representing capability of capturing spatial dependencies due to the fixed graph structure. To adaptively learn the non-linear correlations between nodes, we employ embedding technology and similarity measurement to construct graph adjacency matrices, motivated by the graph structure learning [33]. We first randomly initialize vector representations for nodes, then measure the similarity between nodes' vectors to generate a graph adjacency matrix. The node representations are optimized based on the performance of the downstream task. Specifically, AGL can be formulated as follows,

$$\mathcal{F} = tanh(\alpha \cdot X), \tag{2}$$

$$\mathcal{M}_1 = \mathcal{F}(E_1\mathcal{W}_1), \mathcal{M}_2 = \mathcal{F}(E_2\mathcal{W}_2), \tag{3}$$

$$\mathcal{M} = \mathcal{M}_1\mathcal{M}_2^T - \mathcal{M}_2\mathcal{M}_1^T, \tag{4}$$

$$\mathcal{A} = ReLU(\mathcal{F}(\mathcal{M})), \tag{5}$$

where $\mathcal{F}$ is a tanh activation function with a scalar α adjusting the input X to control the saturation rate of the activation function, The trainable node embeddings $E_1 \in \mathbb{R}^{N\times d_\mathcal{G}}$ and $E_2 \in \mathbb{R}^{N\times d_\mathcal{G}}$ are initialized randomly then continuously optimized during training, N and $d_\mathcal{G}$ are the number of nodes and embedding dimensionality. $\mathcal{W}_1$ and $\mathcal{W}_2$ are trainable linear transformation parameters. $\mathcal{M}$ is a skew-symmetric matrix, thus $-\mathcal{M}$ is equal to $\mathcal{M}^T$ and the diagonal values are zero. $\mathcal{A} \in \mathbb{R}^{N\times N}$ turns an asymmetric matrix to represent the uni-direction correlation through the non-negative linear rectification unit.

The global graph and the local graphs can be constructed by Equation (2)-(5), denoted as $\mathcal{A}_G, \mathcal{A}_L^1, \mathcal{A}_L^2, \cdots, \mathcal{A}_L^S$, where the global node embedding $\{E_1^G, E_2^G\} \in \mathbb{R}^{S\times d_\mathcal{G}^G}$, the local node embedding $\{E_1^L, E_2^L\} \in \mathbb{R}^{M\times d_\mathcal{G}^L}$, the global adjacency matrix $\mathcal{A}_G \in \mathbb{R}^{S\times S}$, and the local adjacency matrix $\mathcal{A}_L^i \in \mathbb{R}^{M\times M}, 1 \le i \le S$. Besides, to enable information transfer across different regions, we utilize the local graph representation as the node feature representation within the global graph. Detailed explanations of this module will be provided in Section 4.4.

4.3. Spatio-temporal learning module

In this subsection, we present STL module, which is designed to capture the temporal dependency of meteorological variables and spatial dependency between variables across regions. STL module primarily comprises two components: gated temporal convolution (GTC) and spatial graph convolution (SGC). The details of both components are described as follows.

4.3.1. Gated temporal convolution

GTC component is specifically designed to capture temporal patterns in long sequences and multi-scale time ranges, aligning with the temporal dimension of a hierarchical spatio-temporal graph. While RNNs-based networks are well-suited for modeling sequential features due to their inherent recursive nature, they have limitations as they restrict the current state to depend solely on the previous state, which can result in increased computational costs. To overcome these limitations, Dilated CNN [34] introduces a dilation factor that determines the number of steps skipped in a standard convolution operation. This approach exponentially expands the receptive field of the network, enabling it to effectively handle long-term sequences. Dilated CNN has demonstrated promising performance in tasks such as audio generation. Mathematically, assuming the dilated convolution with k layers of kernel size c and dilation factor exponentially increasing by rate d ($d > 1$) for each layer, the size of receptive field can be represented as follows,

$$RF^k = 1 + (c-1)(d^k - 1)/(d-1). \tag{6}$$

It can capture longer sequences without increasing the scale of model parameters, compared to the canonical convolution.

Furthermore, it is widely recognized that meteorological changes exhibit distinct variation patterns across different time scales. For instance, temperature is influenced by daylight, typically following a rising and falling trend. Within a day, there can also be various trends observed during different time periods. Motivated by the inception network, which serves as the backbone of GoogLeNet [35,33] and achieves competitive performance in computer vision tasks by concatenating outputs from convolutions with different filter kernel sizes, we integrate the concept of dilated inception. By combining Dilated CNN and inception, we adopt dilated inception as the fundamental unit of the gated temporal convolution, enabling effective modeling of long and diverse temporal dependencies. Formally, considering the transformed high-dimensional hidden feature of layer l, denoted as $\mathcal{H}^l$, the output of dilated inception can be defined as follows,

$$h^l_{\mathcal{T}} = concat(f^d_{1\times c_1}(\mathcal{H}^l), f^d_{1\times c_2}(\mathcal{H}^l), \cdots, f^d_{1\times c_n}(\mathcal{H}^l)), \tag{7}$$

where f represents the convolutional operator, d is the dilation factor, n and c_n denote the number of kernel and corresponding kernel size.

Moreover, for controlling the information passing to the next layer, GTC adopts a gated activation unit, which integrates two dilated inception layers as the final temporal output, formulated as follows,

$$\mathcal{H}^l_{\mathcal{T}} = tanh(h^l_{\mathcal{T}}) \odot sigmoid(h^l_{\mathcal{T}}), \tag{8}$$

where tanh and sigmoid are tangent hyperbolic and sigmoid activation functions, respectively. $\odot$ is the Hadamard product, and $\mathcal{H}^l_{\mathcal{T}}$ is the output of temporal convolution of lth layer. We denote the temporal features of the local spatio-temporal graph and global spatio-temporal graph as $\mathcal{H}^l_{L_j,\mathcal{T}}$ and $\mathcal{H}^l_{G,\mathcal{T}}$, respectively.

4.3.2. Spatial graph convolution

In order to capture the underlying spatial features between meteorological variables and regions, we employ SGC to effectively model these spatial features. Through adaptive learning of local and global graphs, SGC enables the extraction of important spatial information. The graph convolution operation facilitates the learning of node representations by aggregating features from neighboring nodes. First, we present the mathematical form of the graph convolution layer, where its depth-wise propagation rule can be defined as follows,

$$L_{rm} = \widetilde{D}^{-1}\widetilde{A}, \tag{9}$$

$$\mathcal{H}^{l,(d+1)}_S = \beta\mathcal{H}_{in} + (1-\beta)L_{rm}\mathcal{H}^{l,(d)}_S. \tag{10}$$

Here L_{rm} is the normalized laplacian, $\widetilde{A} = A + I$ is the adjacency matrix with self-connections, I is the identity matrix, $\widetilde{D} = \sum_j \widetilde{A}_{ij}$ is the degree matrix, β is a hyperparameter to keep the ratio from original information, $\mathcal{H}_{in}$ represents the input temporal features of the lth local and global gated temporal convolutions. $\mathcal{H}^{l,(d)}_S$ denotes the features of nodes at dth depth of GCN after spatial information aggregation. To further capture the information diversities between different depths of GCN, we utilize trainable parameter weight $\mathcal{W}$ to filter information, which can be formulated as follows,

$$\mathcal{H}^l_S = \sum_{i=0}^{K} \mathcal{H}^{l,(i)}_S \mathcal{W}^i, \tag{11}$$

where K is the depth of spatial graph convolution. Considering the inflow information and outflow information of each node, the spatial features are represented by combining two graph convolution layers with the learned adjacency matrix and its transpose. Correspondingly, $\mathcal{H}^l_{L,S}$ and $\mathcal{H}^l_{G,S}$ can denote the spatial features of the local and global spatio-temporal graph.

4.4. Dynamic interactive learning module

To capture the interaction of spatio-temporal features between local graphs and the global graph, we introduce DIL module. This module addresses the information passing between the two-level graphs through a fusion process and a diffusion process, facilitating information transfer. DIL module comprises an information fusion layer and an information diffusion layer, as illustrated in Fig. 2. These layers are interleaved with the spatio-temporal learning module of the local graph and the global graph.

4.4.1. Information fusion layer

The initial status of nodes in each local graph is determined by the observations of meteorological variables. However, the nodes in the global graph are virtual and lack explicit observed data. To address this problem, we adopt a simple yet effective strategy by fusing the nodes' information from the corresponding weather station to represent the node's hidden status in the global graph. This fusion is achieved through average pooling, which aggregates the nodes' hidden representations. Given the hidden features output from the spatio-temporal learning module of ith local graph in the lth layer, denoted as $\mathcal{H}^l_{L,i}$, the information fusion of the lth layer of the network can be formulated as follows,

$$\mathcal{H}^l_{G,i} = \frac{1}{M}\sum_{j=1}^{M}\mathcal{H}^l_{L_i,j} \tag{12}$$

$$\mathcal{H}^l_G = concat(\mathcal{H}^l_{G,1:S}), \tag{13}$$

where $\mathcal{H}^l_{G,i}$ denotes the feature of ith node of the global graph, S is the number of nodes of the global graph, i.e., the number of weather stations.

4.4.2. Information diffusion layer

The information diffusion aims to propagate the spatio-temporal features from the global graph to the local graphs. To accomplish this, we begin by expanding the feature dimension of the global graph-level output by inserting a new axis and replicating it for each weather station. Next, we employ a gate mechanism to control the extent of information flow directed towards the local graphs. The information diffusion operation can be mathematically expressed as follows,

$$\mathcal{H}^{l+1} = tanh(\mathcal{H}^l_G \otimes \mathbf{1}M) \odot sigmoid(\mathcal{H}^l_G \otimes \mathbf{1}M), \tag{14}$$

where $\otimes$ is the Kronecker product, $\mathbf{1}M$ denotes a tensor of size $1\times 1\times 1\times M$ filled with ones. H^{l+1} is the input features of the next layer as the updated local graph features.

4.5. Output module

To alleviate the vanishing gradient of the multi-layer network and make the most of information from various stages, we employ the skip connection to integrate the features of the output of all layers. Then, the output module consists of two 1×1 standard convolutional layers, and the number of output channels is set as the forecasting time steps to output prediction with our desired time dimension.

4.6. Learning algorithm

Let Θ represent all the training parameters in HiSTGNN. These parameters are learned by minimizing the following mean absolute error (MAE) loss function between the ground truth $\mathcal{Y}$ and prediction $\hat{\mathcal{Y}}$, which can be written as

$$\underset{\Theta}{argmin}\mathcal{L} = \frac{1}{N\cdot S\cdot M\cdot Q}\sum_{i=1}^{N}\sum_{j=1}^{S}\sum_{m=1}^{M}\sum_{t=P+1}^{Q}|\hat{\mathcal{Y}}_{i,j,m,t} - \mathcal{Y}_{i,j,m,t}| \tag{15}$$

where N, S, M, Q are the number of samples, weather stations, meteorological variables, and future time steps. Furthermore, we summarize the training of HiSTGNN in Algorithm 1.

5. Experiments

In this section, we evaluate *HiSTGNN* on multi-step weather forecasting using three real-world weather datasets from different climates to justify our design solutions. We first introduce the relevant experimental settings, including datasets, evaluation metrics, comparison methods, and hyperparameter settings. Then we present the experimental results and analysis in detail.

5.1. Experimental settings

5.1.1. Datasets

To enable reliable assessments for multi-variable and multi-station weather forecasting, we employ three public real-world competition weather datasets.

Algorithm 1 Training of HiSTGNN Algorithm.

Input: The dataset $\mathcal{D}$, batch size N, the number of layers of HiSTGNN L, the number of weather stations S, the number of meteorological variables M, historical step size P, forecasting horizon size Q, the dimensionality of input feature d.
Output: HiSTGNN model.

initialize model parameters Θ
repeat
  sample a batch ($\mathcal{X} \in \mathbb{R}^{N\times S\times M\times P\times d}, \mathcal{Y} \in \mathbb{R}^{N\times S\times M\times Q}$) from $\mathcal{D}$
  compute $\{\mathcal{A}_L^1, \cdots, \mathcal{A}_L^S\} \in \mathbb{R}^{M\times M}$, $\mathcal{A}_G \in \mathbb{R}^{S\times S}$ ▷ build adjacency matrices for local graphs and the global graph
  $\mathcal{H}^0 = start\_conv(\mathcal{X})$, $\mathcal{H}^{skip} = skip\_conv(\mathcal{X})$ ▷ $\mathcal{H}^0 \in \mathbb{R}^{N\times C_{\text{res}}\times M\times P\times S}$, $\mathcal{H}^{skip} \in \mathbb{R}^{N\times C_{\text{skip}}\times M\times P\times S}$
  for $i \leftarrow 1$ **to** L **do**
    for $j \leftarrow 1$ **to** S **do**
      $\mathcal{H}^i_{L_j,\mathcal{T}} = GTC(\mathcal{H}^{i-1}_{(:,:,:,:,j)})$ ▷ $\mathcal{H}^i_{L_j,\mathcal{T}} \in \mathbb{R}^{N\times C_{\text{gated}}\times M\times(P-RF^i)}$, GTC is the gated temporal convolution, L_j denotes the jth local graph
      $\mathcal{H}^i_{L_j,S} = SGC(\mathcal{H}^i_{L_j,\mathcal{T}}, \mathcal{A}^j_L)$ ▷ $\mathcal{H}^i_{L_j,S} \in \mathbb{R}^{N\times C_{\text{gated}}\times M\times(P-RF^i)}$, SGC is the spatial graph convolution
    $\mathcal{H}^{skip} = \mathcal{H}^{skip} + skip\_conv(concat(\mathcal{H}^i_{L_{1:S},\mathcal{T}}))$ ▷ $concat(\mathcal{H}^i_{L_{1:S},\mathcal{T}}) \in \mathbb{R}^{N\times C_{\text{gated}}\times M\times(P-RF^i)\times S}$
    $\mathcal{H}^i_{G,j} = \frac{1}{M}\sum_{k=1}^{M} \mathcal{H}^i_{L_j,(:,:,k,:)}$ ▷ $\mathcal{H}^i_{G,j} \in \mathbb{R}^{N\times C_{\text{gated}}\times 1\times(P-RF^i)}$, average pooling for information fusion
    $\mathcal{H}^i_G = concat(\mathcal{H}^i_{G,1:S})$ ▷ $\mathcal{H}^i_G \in \mathbb{R}^{N\times C_{\text{gated}}\times S\times(P-RF^i)}$
    $\mathcal{H}^i_{G,\mathcal{T}} = GTC(\mathcal{H}^i_G)$ ▷ $\mathcal{H}^i_{G,\mathcal{T}} \in \mathbb{R}^{N\times C_{\text{gated}}\times S\times(P-RF^i)}$, the temporal features of the global-level graph
    $\mathcal{H}^i_{G,S} = SGC(\mathcal{H}^i_{G,\mathcal{T}}, \mathcal{A}_G)$ ▷ $\mathcal{H}^i_{L_j,S} \in \mathbb{R}^{N\times C_{\text{gated}}\times S\times(P-RF^i)}$, the spatial features of the global-level graph
    $\mathcal{H}^i = tanh(\mathcal{H}^l_G \otimes \mathbf{1}M) \odot sigmoid(\mathcal{H}^i_G \otimes \mathbf{1}M)$ ▷ $\mathcal{H}^i \in \mathbb{R}^{N\times C_{\text{gated}}\times M\times(P-RF^i)\times S}$, gate copy for information diffusion
  $\mathcal{H}^{skip} = \mathcal{H}^{skip} + skip\_conv(\mathcal{H}^L)$ ▷ $\mathcal{H}^{skip} \in \mathbb{R}^{N\times C_{\text{skip}}\times M\times(P-RF^L)\times S}$
  $\mathcal{H}^{skip} = \mathcal{H}^{skip} + skip\_conv(\mathcal{H}^L)$
  $\hat{\mathcal{Y}} = end\_cov(\mathcal{H}^{skip})$ ▷ $\hat{\mathcal{Y}} \in \mathbb{R}^{N\times Q\times M\times 1\times S}$
  $\hat{\mathcal{Y}} = swip\_axis(\hat{\mathcal{Y}})$ ▷ $\hat{\mathcal{Y}} \in \mathbb{R}^{N\times S\times M\times Q}$
  compute $loss = \mathcal{L}(\hat{\mathcal{Y}}, \mathcal{Y})$
  compute the stochastic gradient of Θ according to $loss$
  update Θ
until stopping criteria is met
output the learned HiSTGNN model

- *WD_BJ*[1] *[5]*: The weather dataset is hourly collected from 10 ground automatic weather stations with 9 meteorological variables in Beijing, released by an online competition for daily weather forecasting.[2] It focuses on a set time period forecasting, i.e., from 7:00 of the day to 15:00 of the next day, a total of 33 hours. However, the input of the original individual sample spans from 3:00 intraday to 15:00 (UTC) on the next day. The corresponding ground truth covers the time period from 3:00 of the second day to 15:00 of the third day. To avoid data overlap, we push the output backward by 4 hours, and the input forward by 9 hours. Hence, the input time step length is 28, and the forecasting time step length is set to 33. Following [5], we also choose temperature at 2 meters (t2m), relative humidity at meters (rh2m), and wind speed at 10 meters (w10m) as the target variables and sequentially split the dataset into a training set ranging from Mar. 1st, 2015 to May. 31st, 2018, validation set ranging from Jun. 1st, 2018 to Aug. 28th, 2018, and test set ranging from Aug. 29th, 2018 to Nov. 3rd, 2018, respectively.
- *WD_ISR*[3]*:* The weather dataset is hourly collected from OpenWeather[4] ranging from Feb. 2nd, 2012 to Oct. 28th, 2017, and contains 4 weather conditions with temperature, humidity, wind speed, and atmospheric pressure, observed in 6 cities of Israel, including Beersheba, Tel Aviv District, Eilat, Haifa, Nahariyya, and Jerusalem. All 4 meteorological variables are used for forecasting targets. We adopt the same daily forecasting as the study [5] i.e., using a 24-step sliding window. The input time step length is empirically set to 48 hours. The forecasting time step length is 24 hours. 80 percent of data are used for training, 10 percent of data are used for validation while the remaining for testing in chronological order.
- *WD_USA*[3]*:* Except that the dataset is observed in 13 cities in the United States of America, including Boston, New York, Philadelphia, Detroit, Pittsburgh, Chicago, Indianapolis, Charlotte, Saint Louis, Nashville, Atlanta, Jacksonville, and Miami, the other setup of data is identical to WD_ISR.

In all those three datasets, we employ linear interpolation along the temporal dimension to handle missing values, and apply min-max normalization to scale each variable within the range of [0, 1]. Table 2 summarizes the statistics of three datasets.

5.1.2. Evaluation metrics

We measure our method and baselines by three common deviation-based evaluation metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE), which are formulated as follows,

$$RMSE = \sqrt{\frac{1}{n}\sum_i (y_i - \hat{y}_i)^2}, \tag{16}$$

[1] https://github.com/BruceBinBoxing/Deep_Learning_Weather_Forecasting.
[2] AI Challenger 2018 https://challenger.ai/competition/wf2018. The official website is currently not working, but the data is available in the aforementioned GitHub repository.
[3] https://www.kaggle.com/selfishgene/historical-hourly-weather-data.
[4] https://home.openweathermap.org/.

Table 2
Data statistics.

Data	WD_BJ	WD_ISR	WD_USA
Location	Beijing	Iseral	United States of America
Time span	3/1/2015-11/03/2018	10/2/2012-10/28/2017	10/2/2012-10/28/2017
Time interval	1 hour	1 hour	1 hour
Meteorological variable	9	4	4
Weather station	10	6	13
Sample size	1301	1850	1850
Input length (P)	28	48	48
Output length (Q)	33	24	24

$$MAE = \frac{1}{n}\sum_{i}|y_i - \hat{y}_i|, \tag{17}$$

$$MAPE = \frac{1}{n}\sum_{i}|\frac{y_i - \hat{y}_i}{y_i}|, \tag{18}$$

where y and $\hat{y}$ are the ground truth and the predicted values; n is the number of all available predicted values. For them, lower values are better. As same with DUQ [5], we first calculate the performance of each target variable over all time steps, then take the average of the corresponding one as the ultimate RMSE, MAE, and MAPE criteria. Notably, for MAPE, we mask samples with a value of 0 to avoid the issue of division by zero.

5.1.3. Baselines

We compare our HiSTGNN with the following 7 baselines:

- SARIMA [15]: Seasonal Autoregressive Integrated Moving Average is a classic statistical univariate time series forecasting method, which utilizes the autoregressive, differencing, moving average, and all three seasonal components to estimate future values, where its parameters are chosen based on the AIC (Akaike information criterion).
- SVR [36]: Support Vector Regression is a non-linear variant of support vector machine for regression, which is widely used for covariate time series tasks.
- Seq2Seq [37]: Sequence to Sequence model employs an encoder-decoder architecture to effectively model temporal dependencies for time series forecasting. It is built upon a GRU-based network with two layers, each consisting of 300 hidden units.
- WaveNet [38]: A deep generative model for generating speech that employs dilated causal convolution with large receptive fields to handle long-range temporal dependencies.
- DUQ [5]: A deep uncertainty quantification method that simultaneously outputs the mean and variance estimations for weather forecasting. It is two layers GRU-based seq2seq with 300 hidden nodes of each layer.
- AGCRN [39]: A spatio-temporal forecasting method that combines graph convolutional network and recurrent neural network to capture spatio-temporal dependencies.
- MTGNN [33]: A state-of-the-art spatio-temporal model for multivariate time series forecasting that utilizes dilated inception and graph convolution to discover the long-term temporal patterns and the uni-directed spatial relations among variables.

5.1.4. Implementation details

All the deep learning model training experiments are conducted on Nvidia Titan RTX GPUs, and implemented by PyTorch of version 1.2.0 and Python of version 3.6, except for DUQ and Seq2Seq, which are implemented by TensorFlow following their source codes. The source code is available at https://github.com/mb-Ma/HiSTGNN.

HiSTGNN. For all datasets, the model is trained by the Adam optimizer with gradient clip 5. The learning rate is set to 0.001. The L2 regularization penalty is 0.0001. The epoch is set to 100. Early stopping with 15 patience is used to select optimal model parameters, i.e., the training is stopped when the performance of the validation set has not improved 15 times. Following [33], The temporal convolution and graph convolution both have 32 output channels. The skip connection layers all have 64 output channels. The first convolution and second convolution of the output module have 128 and 1 output channels. The depth of the graph convolution module is set to 2. The β is set to 0.05. We vary five core hyper-parameters, including the batch size among {16, 32, 64}, the depth of the network among {1, 2, 3, 4, 5}, the saturation rate {1, 2, 3, 4}, and the embedding dimensionality of variable and station node {5, 10, 20, 30}, to determine the optimal parameter combination and assess the sensitivity of the model (More details will be provided in Section 5.6). Instead of utilizing grid search to obtain the optimal performance of all parameter combinations, which costs massive computing resources, we sequentially determine the optimal parameters, suggesting that better performance improvements are possible. Ultimately, for the WD_BJ dataset, the WD_ISR dataset, and WD_USA, the batch size is (64, 16, and 32). The depth of the network is (3, 2, and 2), the saturation rate is (3, 2, and 2), and the embedding dimensionalities of the variable and station are (20, 5, and 5) and (20, 10, and 10).

Baselines. Considering SARIMA as a univariate time series forecasting method, we train a separate model for each variable of each weather station. The orders are automatically searched using the *pmdarima* python package, where the p, q, and d range from 1 to 5; the seasonal P, Q, and D also range from 1 to 5; the period for seasonal differencing is set to 12. For SVR, we design a covariate time series forecasting approach where the historical observations of the target variable and other weather variables are

Table 3
Performance comparisons on WD_BJ dataset.

Methods	Metrics	TEMP	HUM	WIND	Avg.
SARIMA	MAE	6.1973	21.8582	1.2860	9.7805
	RMSE	7.7761	27.0832	1.7874	12.2156
	MAPE (%)	40.06	35.09	71.22	48.79
SVR	MAE	5.6790	20.1549	2.4524	9.4288
	RMSE	6.4088	22.7126	2.6346	10.5853
	MAPE (%)	37.92	29.84	63.23	43.66
Seq2Seq	MAE	2.0259	10.8049	0.8954	4.5754
	RMSE	2.6985	14.7942	1.4274	6.3068
	MAPE (%)	36.11	26.52	61.80	41.47
WaveNet	MAE	2.2591	10.9460	1.0051	4.7368
	RMSE	2.9978	15.1380	1.3396	6.4918
	MAPE (%)	38.96	25.70	74.70	46.45
DUQ	MAE	2.0221	10.2879	0.8801	4.3967
	RMSE	2.7343	14.9173	1.2702	6.3072
	MAPE (%)	34.23	25.88	53.51	37.87
MTGNN	MAE	2.0181	10.3525	0.8684	4.4130
	RMSE	2.7186	14.9944	1.2596	6.3242
	MAPE (%)	36.72	25.44	54.98	39.05
AGCRN	MAE	**1.8912**	10.5406	0.8932	4.4416
	RMSE	**2.6270**	14.9901	**1.2576**	6.2916
	MAPE (%)	31.10	26.09	57.83	38.34
HiSTGNN(our)	MAE	1.9533	9.8089	**0.8671**	**4.2098**
	RMSE	2.6353	**14.0129**	1.2634	**5.9705**
	MAPE (%)	**32.49**	**23.80**	**52.40**	**36.23**

used as input to predict future values of the target variable. We employ SVR with the RBF (radial basis function) kernel, setting the penalty term C to 0.1 and the epsilon value to 0.2 for all datasets. Following [5], we utilize a two-layer GRU network with 300 hidden units as both the encoding and decoding layers for the Seq2Seq and DUQ models. The Seq2Seq utilizes mean squared error (MSE) as the loss function, while DUQ adopts negative log-likelihood estimation (NLE) as the loss function. Regarding WaveNet, we set the dilation factor to 2, the stack size to 6, and the number of residual convolution channels and skip convolution channels to be the same as those in HiSTGNN. In the spatio-temporal forecasting model, following [10], we treat the weather stations as nodes of a graph that represent the meteorological associations between regions. In addition to the node dimensionality which is the same as the dimensionality of the weather station in HiSTGNN, the remaining parameters are set according to the original paper. Except for SARIMA and SVR, all models are trained by the Adam optimizer.

5.2. Main results

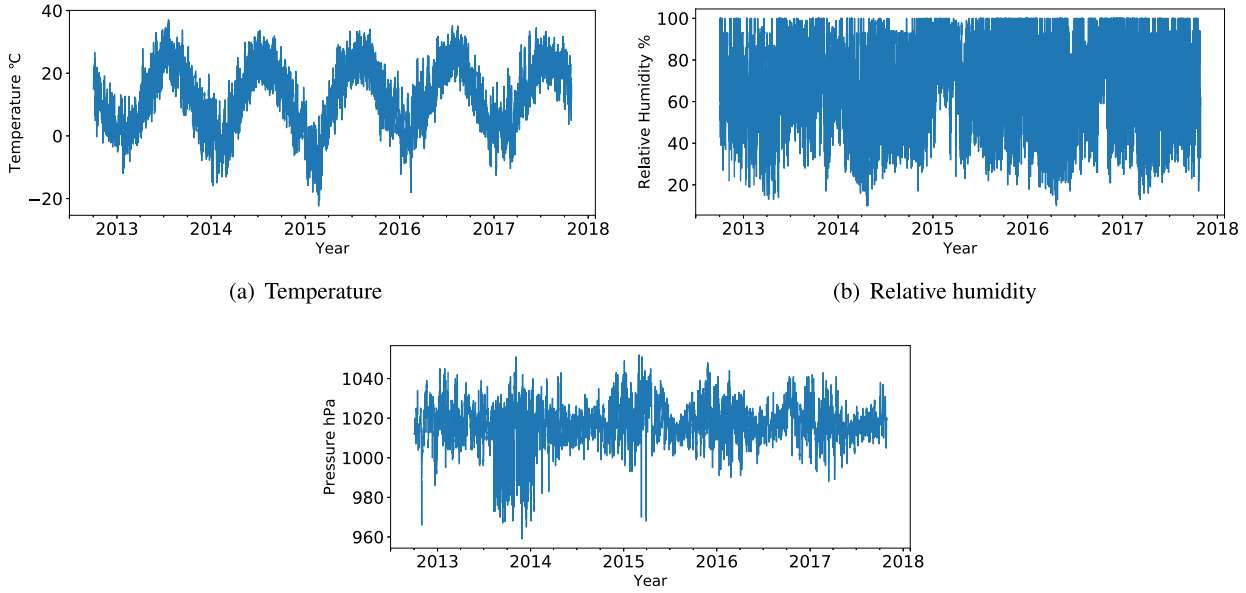
Table 3, Table 4, and Table 5 present the experimental results of HiSTGNN and compared methods on WD_BJ, WD_ISR, and WD_USA datasets, respectively. These tables provide insights into the performance metrics including MAE, RMSE, and MAPE for each variable, as well as the average performance across all variables. We can observe that the spatio-temporal models, which consider the spatial correlation between variables, outperform the time series forecasting models. In particular, HiSTGNN achieves remarkable improvements on the WD_BJ dataset, surpassing the state-of-the-art weather forecasting method DUQ with reductions of 4.25% in MAE and 5.34% in RMSE. This improvement can be attributed to HiSTGNN's effective handling of spatial features, as the target variables with weak or no seasonal variation are challenging to capture using solely temporal features. For example, when examining the temperature, relative humidity, and pressure of New York in the WD_USA dataset (see Fig. 3), it becomes evident that only the temperature exhibits a strong seasonal variation, while the relative humidity and pressure are subject to significant noise. In general, HiSTGNN achieves new state-of-the-art performance on the majority of the evaluation items, with a success ratio of 30 out of 42 (considering each metric for each variable of each dataset as an item). Notably, with the exception of the MAPE metric on the WD_ISR dataset, HiSTGNN outperforms all baselines in terms of the average performance across meteorological variables on all datasets. This justifies our design choices in simultaneously learning the associations among multiple meteorological variables. In comparison, MTGNN and AGCRN, which employ flat graph structures to consider the correlation of meteorological variables between regions, fall short in handling the transformation between meteorological variables, leading to inferior performance compared to HiSTGNN.

Table 4
Performance comparisons on WD_ISR dataset.

Methods	Metrics	TEMP	HUM	WIND	PSUR	Avg.
SARIMA	MAE	3.1419	15.8560	1.3799	3.6863	6.0160
	RMSE	4.0347	19.2929	1.7756	6.2767	7.8450
	MAPE (%)	12.98	30.76	61.85	0.36	26.49
SVR	MAE	2.927	9.3751	1.0237	3.427	4.1882
	RMSE	3.6021	13.6427	1.5296	5.9853	6.1899
	MAPE (%)	6.32	15.46	36.26	1.28	14.83
Seq2Seq	MAE	1.3726	7.9663	0.9314	2.7346	3.2512
	RMSE	1.8923	11.9565	1.4136	5.3360	5.1496
	MAPE (%)	5.51	15.67	36.25	0.27	14.42
WaveNet	MAE	1.5107	7.4604	0.9231	2.7631	3.1796
	RMSE	1.9306	11.2613	1.3683	4.2814	4.7104
	MAPE (%)	5.81	14.57	36.84	0.27	14.37
DUQ	MAE	1.3487	8.0896	0.9213	3.0548	3.3536
	RMSE	1.8748	11.2739	1.3186	5.1690	4.9091
	MAPE (%)	5.35	16.53	36.25	0.30	14.61
MTGNN	MAE	1.4483	7.5238	1.0138	2.5149	3.1252
	RMSE	1.9290	11.2421	1.4185	4.2898	4.7198
	MAPE (%)	5.69	14.87	**35.13**	0.25	**13.99**
AGCRN	MAE	1.2644	7.5966	0.9105	2.5471	3.0770
	RMSE	1.7461	11.3350	**1.2828**	4.6245	4.7471
	MAPE (%)	5.07	15.37	38.00	0.25	14.67
HiSTGNN(our)	MAE	**1.2551**	**7.2302**	**0.9018**	**2.3911**	**2.9446**
	RMSE	**1.7287**	**10.9434**	1.3038	**4.2666**	**4.5606**
	MAPE (%)	**4.98**	**14.66**	36.73	**0.23**	14.15

Table 5
Performance comparisons on WD_USA dataset.

Methods	Metrics	TEMP	HUM	WIND	PSUR	Avg.
SARIMA	MAE	2.3112	11.6933	1.4677	3.2208	4.6732
	RMSE	3.2113	15.9734	2.0920	4.4713	6.4370
	MAPE (%)	13.75	18.09	62.75	0.31	23.73
SVR	MAE	2.0571	10.2348	1.3308	3.0054	4.1570
	RMSE	3.0018	13.9286	1.9766	4.8748	5.9455
	MAPE (%)	11.11	14.73	53.16	0.55	19.89
Seq2Seq	MAE	1.8765	9.3221	1.1797	2.7622	3.7851
	RMSE	2.5627	12.2728	1.7194	4.1885	5.1858
	MAPE (%)	10.23	14.40	52.74	0.20	19.39
WaveNet	MAE	1.9139	9.4432	1.2654	2.4281	3.7626
	RMSE	2.6709	12.5661	1.6851	3.1876	5.0274
	MAPE (%)	11.57	14.54	61.45	0.20	21.94
DUQ	MAE	1.9424	9.8207	**1.1663**	2.4250	3.8386
	RMSE	2.6363	12.4793	**1.5862**	3.4099	5.0279
	MAPE (%)	10.62	15.93	**52.15**	0.23	19.73
MTGNN	MAE	1.8201	9.1751	1.1749	2.2735	3.6109
	RMSE	2.4732	12.3581	1.5930	3.0508	4.8687
	MAPE (%)	**10.06**	13.97	53.71	0.22	19.49
AGCRN	MAE	1.9040	9.2218	1.2746	2.1384	3.6347
	RMSE	2.5505	12.3767	1.8039	3.0105	4.9354
	MAPE (%)	10.59	14.15	53.11	**0.19**	19.51
HiSTGNN(our)	MAE	**1.7546**	**9.1397**	1.1760	**2.0975**	**3.5419**
	RMSE	**2.4247**	**12.2222**	1.6120	**2.9052**	**4.7910**
	MAPE (%)	10.13	**13.88**	52.91	0.20	**19.28**



(a) Temperature (b) Relative humidity

(c) Pressure

Fig. 3. Visualization on three target variables in New York from 02/10/2012-10/28/2017.

Table 6
RMSE and MAE on WD_BJ test set using HiSTGNN with different types of graphs.

Methods	TEMP		HUM		WIND		Avg.	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
w/o LGraph	2.1456	2.8634	10.4695	14.9019	0.8514	1.2521	4.4889	6.3391
w/o GGraph	1.9777	2.7047	10.2911	14.9029	0.8969	1.3236	4.3886	6.3105

Table 7
Comparison on WD_BJ test set with variants of dynamic interaction.

Interaction type	Fusion	Diffusion	MAE	RMSE
DI	Avg Pool	Gate	**4.2098±0.0568**	**5.9705±0.0769**
DI	Max Pool	Gate	4.3743±0.0997	6.0537±0.1244
DI	Avg Pool	w/o	4.3545±0.0812	6.0268±0.1061
DI	Max Pool	w/o	4.3641±0.0923	6.0238±0.1079
w/o DI	–	–	4.5321±0.0587	6.2003±0.0602
OSI	Avg Pool	Gate	4.4351±0.0524	6.1520±0.0771
OSI	Max Pool	Gate	4.4371±0.0426	6.1514±0.0437
OSI	Avg Pool	w/o	4.4729±0.0515	6.1872±0.0678
OSI	Max Pool	w/o	4.4404±0.0390	6.1707±0.0821

5.3. Study of the hierarchical graph

As the hierarchical graph includes the local-level graph and the global-level graph, we validate our hierarchical graph by conducting separate experiments using only local graphs and only global graphs. We refer to HiSTGNN without specific components as follows:

- w/o LGraph: HiSTGNN without the local graph modeling between meteorological variables. We remove the local graph convolution.
- w/o GGraph: HiSTGNN without the global graph modeling between weather stations. In this variant, we remove the global graph convolution and train HiSTGNN using data that is flattened by the station dimension.

Table 6 presents the performance of HiSTGNN with either the global or local graph, both demonstrating the performance degradation compared to hierarchical graphs. We also find that temperature and relative humidity exhibit improved performance when

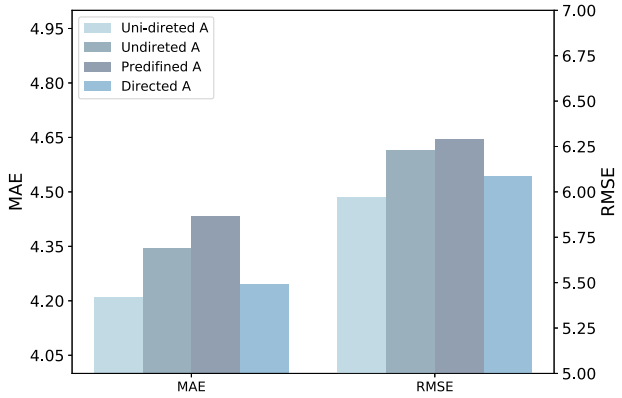


Fig. 4. Performances on four types of graph adjacency matrix in WD_BJ test set.

modeled on local graphs, while wind speed is better suited for cross-regional correlation modeling. This inspires us to focus on the specific modeling of meteorological features in future research.

5.4. Study of adaptive graph learning

To demonstrate the effectiveness of adaptive graph learning, we compare three methods for constructing the graph adjacency matrix A: 1). Predefined graph A, calculated using the correlation coefficient matrix; 2). Undirected graph A, computed based on the similarity of node embeddings, represented as $A = softmax(ReLU(EE^T))$; 3). Directed graph A, similar to the undirected graph but considering two node embeddings, denoted as $A = softmax(ReLU(E_1 E_2^T))$. Fig. 4 illustrates the results obtained. The findings clearly indicate that the uni-directed graph adjacency achieves the lowest MAE and RMSE, demonstrating significant superiority over both the predefined graph and the undirected graph. Although the directed graph with the same structure performs slightly better, particularly in terms of MAE.

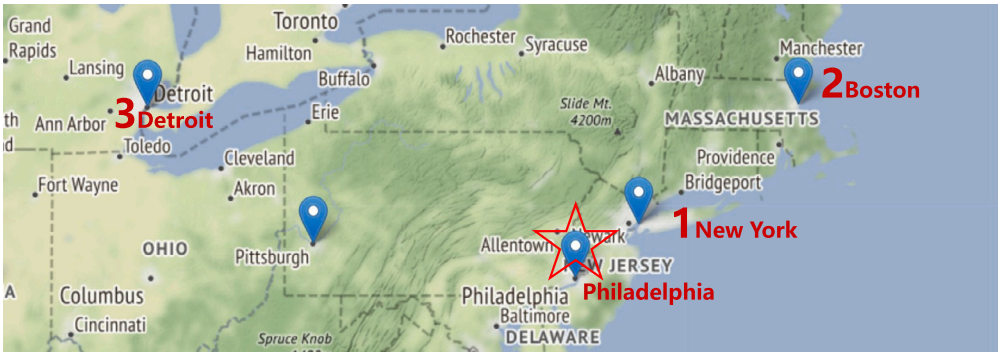
To further examine the effectiveness of the learned local graphs and the global graph, we conducted a case study using the WD_USA dataset. In Fig. 5(a), we visualize the weather stations of the WD_USA dataset and highlight Philadelphia, along with its three neighboring stations: New York, Boston, and Detroit. These stations were selected based on their highest weights, indicating strong correlations. Philadelphia and New York, sharing similar geographical and temperate continental climates, exhibit the highest correlation. Similarly, despite their spatial distance, Boston and Philadelphia are both influenced by ocean currents and situated in the same climatic zone. Fig. 5(b) illustrates the temperature changes from September 8th to September 12th, 2013, for the four highlighted cities. Although there are gaps among the temperature curves, they display a similar trend, highlighting the temporal similarities in temperature patterns across the cities. Furthermore, in Fig. 5(c), we present a heatmap representing the mean edge weights among variables for the 13 weather stations. This heatmap reveals the relationships between relative humidity and other meteorological factors such as temperature, atmospheric pressure, and wind speed. Consistent with meteorological theory, the relationship between relative humidity and these factors sequentially decreases. It is important to note that the correlation coefficients are relatively small due to the complexity of the associations among meteorological factors.

5.5. Study of the dynamic interactive learning

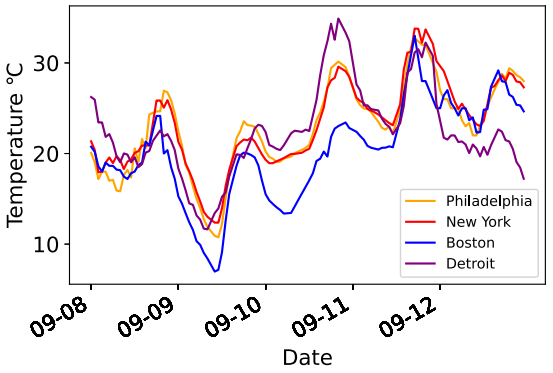
In this section, we conduct experiments on various variants of HiSTGNN to assess the effectiveness of the proposed dynamic interactive learning module. This module plays a crucial role in establishing the message-passing mechanism between different weather stations, utilizing the information fusion and diffusion mechanisms described in Section 4.4. There are three aspects that need to note: 1). interaction type, including the following ways:

- **DI**: HiSTGNN following iterative interaction with the multi-layer stacking network.
- **OSI**: HiSTGNN with one-shot interaction where we replace the interaction between local graphs and the global graph per layer with only one fusion and diffusion operation at the beginning and the end.
- **w/o DI**: HiSTGNN without hierarchical graph structure in which we remove the related components of the global graph on temporal and spatial learning.

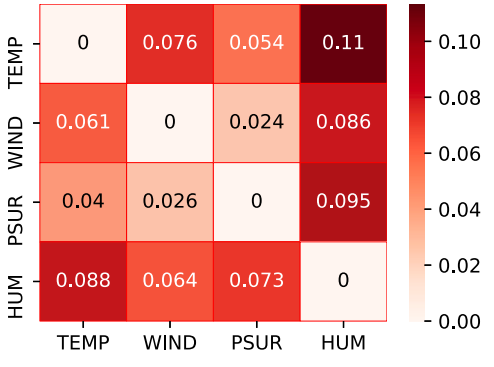
2). fusion with average pooling or max pooling. 3) diffusion with gate mechanism or w/o. Hence, there are 9 variants in total as shown in Table 7. We repeat per experiment 10 times with the same parameters for all variants and report the average of MAE and RMSE and the standard deviation. From the table, we can find that *dynamic interaction* + *average pooling* + *gate* outperforms other variants. In detail, DL is better than OSI among the three interaction types. OSI is superior to w/o DI whereas the latter is still competitive compared to baselines. Besides, the variants of diffusion have an outstanding impact on performance than ones of fusion.



(a) The Philadelphia node with its top-3 neighbors in adjacency matrix of adaptive graph learning



(b) Temperature trends of four cities from 8th Sep. to 12th Sep. 2013



(c) The learned edge weights among variables on WD_USA test set

Fig. 5. A case study of the correlations between variables.

A reason may be that the gate mechanism has more effect on information propagation than the diversity between average pooling and max pooling.

5.6. Study of model parameters

5.6.1. Effect of graph size

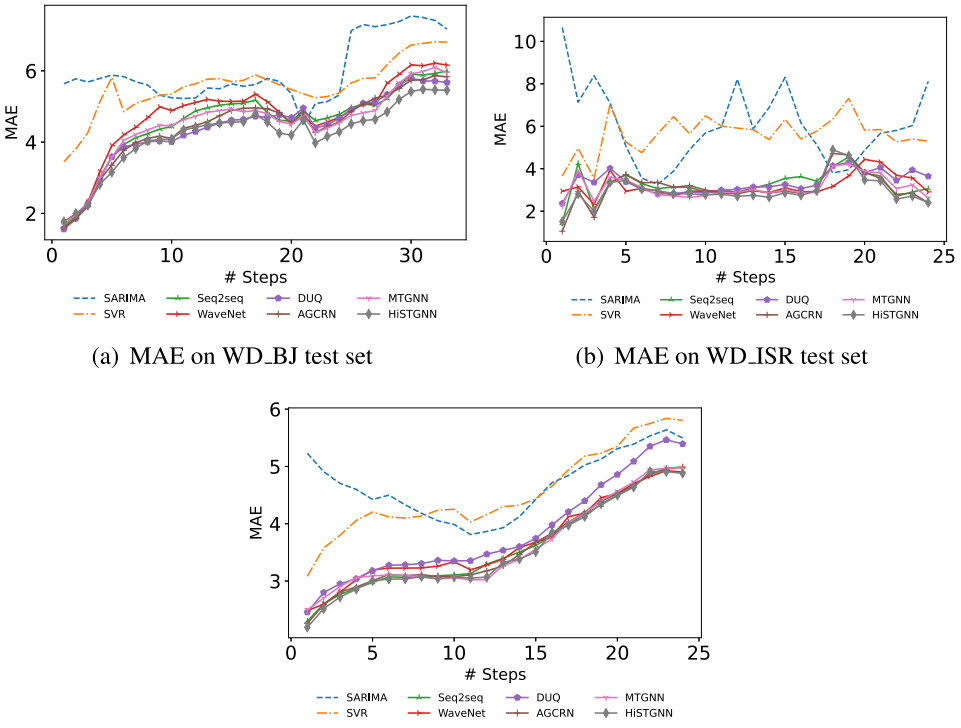
To further explore the impact of graph scale on the weather forecasting model, we varied the number of meteorological variables from the WD_BJ dataset, selecting [3, 4, 6, 8, 9] variables. For consistency, we used a fixed number of 10 weather stations across all cases. As depicted in Fig. 7(a), the visualization of the results demonstrates a gradual improvement in prediction performance as the variable-level graph expands. However, it is important to acknowledge that the exploration of scale boundaries for the variable-level graph is limited by the available number of meteorological variables. Regarding the number of weather stations, we did not conduct experiments with varying numbers due to inherent variations in prediction performance among stations. Therefore, the final results would be influenced not only by the capacity of global graph learning with different sizes but also by the inherent bias among stations.

5.6.2. Effect of network depth

Fig. 7(b) provides an analysis of the impact of network depth on the WD_BJ dataset. Increasing the depth of the network architecture enables an expanded spatio-temporal receptive field, thereby enhancing the model's representation capability. However, it is crucial to acknowledge that as the network depth becomes significantly deeper, the training process becomes more challenging, resulting in a gradual increase in the mean absolute error (MAE).

5.7. Performance on multi-time step forecasting

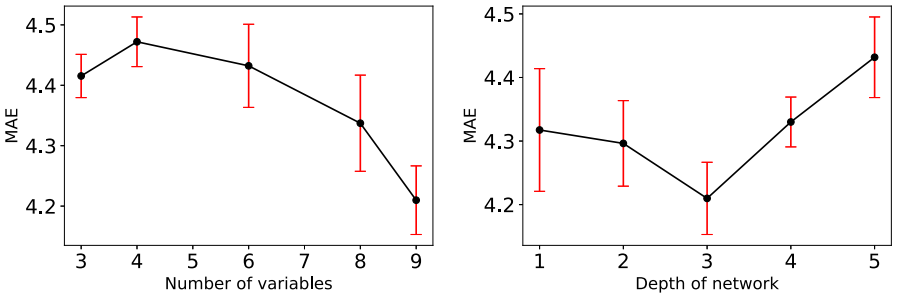
In this section, we conducted an in-depth analysis of the multi-time step forecasting capability of HiSTGNN and the baseline models. Fig. 6 presents a performance comparison, highlighting the consistent superiority of neural network-based methods over statistical machine learning methods. Notably, HiSTGNN consistently outperformed all other models across all datasets and most of



(a) MAE on WD_BJ test set (b) MAE on WD_ISR test set

(c) MAE on WD_USA test set

Fig. 6. MAE on three weather datasets with baselines over multiple time steps.



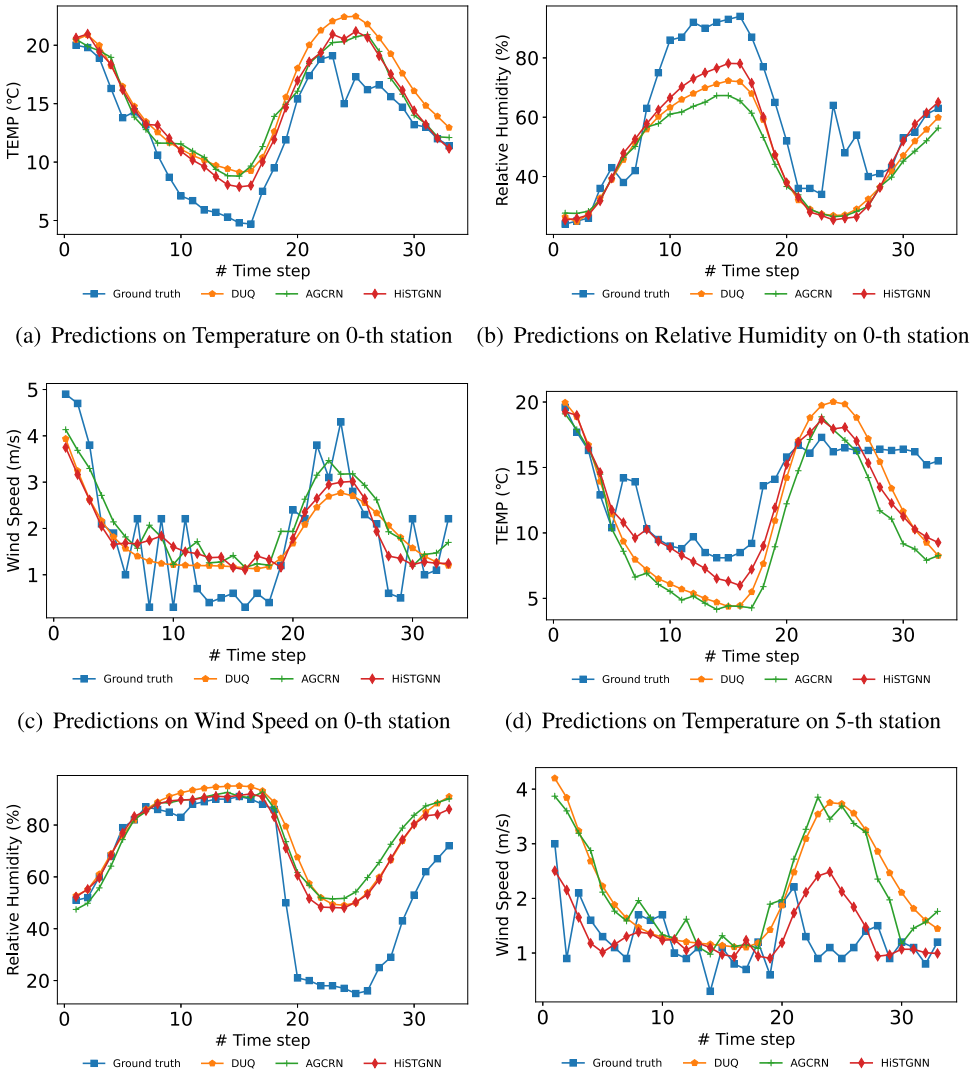
(a) Number of variables (b) Depth of network

Fig. 7. Effect of graph size and network depth on WD_BJ data.

the multiple-time steps, as indicated by the diamond-shaped gray line surpassing the other lines. It is worth noting that the MAE gradually increases with each successive time step, in line with the principles of multi-time step forecasting. As time progresses, the uncertainty in the forecasts tends to grow (as depicted in Fig. 6(a)). However, it is also important to acknowledge that predictions may be more accurate at specific points in time due to the periodic changes in time series data (as shown in Figs. 6(b) and 6(c)).

5.8. Case study

To gain further insights into the forecasting performance, we conducted visualizations comparing the ground truths of different meteorological variables with the predictions generated by our proposed method (HiSTGNN) and two competitive methods (DUQ and AGCRN) on WD_BJ test set. The visualizations, depicted in Fig. 8, illustrate that all three methods are capable of capturing the general trends of temperature, relative humidity, and wind speed. HiSTGNN performs relatively closer to the ground truth, particularly at the peak values. However, a closer examination of Fig. 8(f) reveals that wind speed exhibits more fluctuations, and all three methods struggle to accurately predict the downward trend. This observation highlights the challenge of capturing local time-scale variations while avoiding prediction instability caused by an excessive focus on local dynamics. Addressing this challenge will be an important aspect of our future research.



(a) Predictions on Temperature on 0-th station (b) Predictions on Relative Humidity on 0-th station

(c) Predictions on Wind Speed on 0-th station (d) Predictions on Temperature on 5-th station

(e) Predictions on Relative Humidity on 5-th station (f) Predictions on Wind Speed on 5-th station

Fig. 8. Visualization of predictions of three methods and the ground truth for the 30th sample of WD_BJ test set.

6. Conclusion

In this paper, we proposed a hierarchical spatio-temporal graph neural network (HiSTGNN) to capture spatio-temporal dependencies between meteorological variables across multiple weather stations in an end-to-end way. HiSTGNN first constructs adjusted dependency matrices of the local graphs and the global graph using an adaptive graph learning module. Subsequently, it employs spatio-temporal learning modules to model spatial and temporal dependencies. Additionally, a dynamic interactive learning module is designed to aggregate the representation of the local graph into the corresponding global node and propagate information in the opposite direction. Experimental results demonstrate that our proposed method achieves state-of-the-art performance on three real-world meteorological datasets.

Despite the promising results obtained with our proposed method, there are several limitations that need to be addressed. First, there exists an unstable learning process when modeling multiple sites and variables simultaneously. That is, although the validation loss shows a decreasing trend, there are observable fluctuations, which may be caused by the challenge of finding the optimal solution in multi-meteorological variables optimization and the relatively small data. Second, when dealing with the meteorological data collected from large-scale weather stations, the hierarchical graph's scale increases significantly, posing significant challenges to computational efficiency. In future research, we will conduct further investigations on multi-variable optimization and attempt to solve large-scale weather station data using graph sampling and parallelization techniques.

CRediT authorship contribution statement

Minbo Ma: Conceptualization, Investigation, Methodology, Software, Writing – original draft. **Peng Xie:** Data curation, Formal analysis, Funding acquisition. **Fei Teng:** Data curation, Formal analysis, Software. **Bin Wang:** Data curation, Formal analysis, Software. **Shenggong Ji:** Data curation, Formal analysis. **Junbo Zhang:** Data curation, Formal analysis. **Tianrui Li:** Funding acquisition, Project administration, Resources, Supervision, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 62176221).

References

[1] T. Gneiting, A.E. Raftery, Weather forecasting with ensemble methods, Science 310 (5746) (2005) 248–249.
[2] J. Burton, Robert fitzroy and the early history of the meteorological office, Br. J. Hist. Sci. 19 (2) (1986) 147–176.
[3] M. Tolstykh, A. Frolov, Some current problems in numerical weather prediction, Izv., Atmos. Ocean. Phys. 41 (3) (2005) 285–295.
[4] X. Ren, X. Li, K. Ren, J. Song, Z. Xu, K. Deng, X. Wang, Deep learning-based weather prediction: a survey, Big Data Res. 23 (2021) 100178.
[5] B. Wang, J. Lu, Z. Yan, H. Luo, T. Li, Y. Zheng, G. Zhang, Deep uncertainty quantification: a machine learning approach for weather forecasting, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, 2019, pp. 2087–2095.
[6] P. Hewage, A. Behera, M. Trovati, E. Pereira, M. Ghahremani, F. Palmieri, Y. Liu, Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station, Soft Comput. 24 (21) (2020) 16453–16482.
[7] S. Mehrkanoon, Deep shared representation learning for weather elements forecasting, Knowl.-Based Syst. 179 (2019) 120–128.
[8] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, in: Conference Track Proceedings, 2015.
[9] M. Defferrard, M. Milani, F. Gusset, N. Perraudin, DeepSphere: a graph-based spherical CNN, in: International Conference on Learning Representations (ICLR), 2020.
[10] H. Lin, Z. Gao, Y. Xu, L. Wu, L. Li, S.Z. Li, Conditional local convolution for spatio-temporal meteorological forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 7470–7478.
[11] Z. Yu, S. Miller, F. Montalto, U. Lall, The bridge between precipitation and temperature–pressure change events: modeling future non-stationary precipitation, J. Hydrol. 562 (2018) 346–357.
[12] P. Bauer, A. Thorpe, G. Brunet, The quiet revolution of numerical weather prediction, Nature 525 (7567) (2015) 47–55.
[13] S. Ashkboos, L. Huang, N. Dryden, T. Ben-Nun, P. Dueben, L. Gianinazzi, L. Kummer, T. Hoefler, Ens-10: A dataset for post-processing ensemble weather forecasts, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Advances in Neural Information Processing Systems, vol. 35, Curran Associates, Inc., 2022, pp. 21974–21987.
[14] T. Kurth, S. Subramanian, P. Harrington, J. Pathak, M. Mardani, D. Hall, A. Miele, K. Kashinath, A. Anandkumar, Fourcastnet: accelerating global high-resolution weather forecasting using adaptive Fourier neural operators, in: Proceedings of the Platform for Advanced Scientific Computing Conference, PASC '23, Association for Computing Machinery, New York, NY, USA, 2023.
[15] L. Chen, X. Lai, Comparison between arima and ann models used in short-term wind speed forecasting, in: 2011 Asia-Pacific Power and Energy Engineering Conference, IEEE, 2011, pp. 1–4.
[16] M. Tektaş, Weather forecasting using anfis and arima models, Environ. Res. Eng. Manage. 51 (1) (2010) 5–10.
[17] N.I. Sapankevych, R. Sankar, Time series prediction using support vector machines: a survey, IEEE Comput. Intell. Mag. 4 (2) (2009) 24–38.
[18] P.R.P.G. Hewage, M. Trovati, E.G. Pereira, A. Behera, Deep learning-based effective fine-grained weather forecasting model, Pattern Anal. Appl. 24 (2020) 343–366.
[19] A. Grover, A. Kapoor, E. Horvitz, A deep hybrid model for weather forecasting, in: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 379–386.
[20] G.E. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (2006) 1527–1554.
[21] Y. Han, L. Mi, L. Shen, C. Cai, Y. Liu, K. Li, G. Xu, A short-term wind speed prediction method utilizing novel hybrid deep learning algorithms to correct numerical weather forecasting, Appl. Energy 312 (2022) 118777.
[22] X. Zhang, C. Huang, Y. Xu, L. Xia, P. Dai, L. Bo, J. Zhang, Y. Zheng, Traffic flow forecasting with spatial-temporal graph diffusion network, in: Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, February 2-9, 2021, 2021, pp. 15008–15015.
[23] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, Z. Li, Deep multi-view spatial-temporal network for taxi demand prediction, in: S.A. McIlraith, K.Q. Weinberger (Eds.), Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI-18, New Orleans, Louisiana, USA, February 2-7, 2018, 2018, pp. 2588–2595.
[24] A. Jain, A.R. Zamir, S. Savarese, A. Saxena, Structural-rnn: deep learning on spatio-temporal graphs, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016, IEEE Computer Society, 2016, pp. 5308–5317.
[25] C. Wang, Y. Zhu, T. Zang, H. Liu, J. Yu, Modeling inter-station relationships with attentive temporal graph convolutional network for air quality prediction, in: L. Lewin-Eytan, D. Carmel, E. Yom-Tov, E. Agichtein, E. Gabrilovich (Eds.), WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, ACM, 2021, pp. 616–634.
[26] Y. Wu, X. Yang, Y. Tang, C. Zhang, G. Zhang, W. Zhang, Inductive spatiotemporal graph convolutional networks for short-term quantitative precipitation forecasting, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–18.
[27] Q. Ni, Y. Wang, Y. Fang, GE-STDGN: a novel spatio-temporal weather prediction model based on graph evolution, Appl. Intell. 52 (7) (2022) 7638–7652.

[28] J. Han, H. Liu, H. Zhu, H. Xiong, D. Dou, Joint air quality and weather prediction based on multi-adversarial spatiotemporal networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 4081–4089.
[29] R. Keisler, Forecasting global weather with graph neural networks, arXiv preprint, arXiv:2202.07575, 2022.
[30] R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, A. Pritzel, S. Ravuri, T. Ewalds, F. Alet, Z. Eaton-Rosen, et al., Graphcast: learning skillful medium-range global weather forecasting, arXiv preprint, arXiv:2212.12794, 2022.
[31] Z. Ying, J. You, C. Morris, X. Ren, W.L. Hamilton, J. Leskovec, Hierarchical graph representation learning with differentiable pooling, in: S. Bengio, H.M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, 2018, pp. 4805–4815.
[32] K. Guo, Y. Hu, Y. Sun, S. Qian, J. Gao, B. Yin, Hierarchical graph convolution networks for traffic forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 151–159.
[33] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, C. Zhang, Connecting the dots: multivariate time series forecasting with graph neural networks, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, Association for Computing Machinery, New York, NY, USA, 2020, pp. 753–763.
[34] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, arXiv preprint, arXiv:1511.07122, 2015.
[35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[36] B. Wolff, J. Kühnert, E. Lorenz, O. Kramer, D. Heinemann, Comparing support vector regression for pv power forecasting to a physical modeling approach using measurement, numerical weather prediction, and cloud motion data, Sol. Energy 135 (2016) 197–208.
[37] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, 2014, pp. 3104–3112.
[38] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A.W. Senior, K. Kavukcuoglu, Wavenet: A generative model for raw audio, in: The 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13–15 September 2016, ISCA, 2016, p. 125.
[39] L. Bai, L. Yao, C. Li, X. Wang, C. Wang, Adaptive graph convolutional recurrent network for traffic forecasting, in: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020, pp. 17804–17815.